Назначение лекции

Вы узнаете, как использовать циклы в Python.

Новые слова

**for –** оператор цикла с известным числом повторения.

**while –** оператор цикла с заранее неизвестным числом повторения.

# Цикл for

Цикл **for** позволяет выполнить фрагмент кода заданное количество раз.

for i in range(1,10):

print(i)

В этом примере будет выведена последовательность чисел

1, 2, 3, 4, 5, 6, 7, 8 и 9.

**Добравшись до 10, цикл останавливается, поэтому 10 в выводе отсутствует.**

В этом цикле переменная с именем **i** используется для управления количеством повторений цикла. Сначала **i** присваивается 1 (начальное значение функции **range**). При каждом повторении цикла переменная **i** увеличивается на 1 и выводится ее текущее значение. Цикл повторяется, пока переменная не достигнет 10, после чего цикл останавливается.

**for i in range(1, 10, 2):**

**print(i)**

Функция **range** включает третье значение, которое определяет приращение **i** при каждом проходе цикла (в данном случае 2). В этом случае будут выведены следующие числа: **1, 3, 5, 7, 9**.

**for i in range(10, 1, -3):**

**print(i)**

В этом диапазоне переменная **i** будет каждый раз уменьшаться на 3. Результат: **10, 7, 4**.

**word=’одна строка’**

**for i in word:**

**print(i)**

Каждый символ в строке с именем **word** будет выведен в отдельной строке.

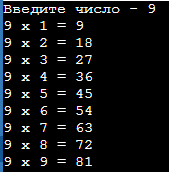
Рассмотрим интересную задачу: пользователь вводит число от 1 до 9. Требуется вывести таблицу умножения для этого числа. Решение очень краткое. Вот оно:

k=int(input("Введите число - "))

for i in range(1,k+1):

print(k, 'x', i, '=', k\*i)

Ответ.



Второй вариант решения:

k=int(input("Введите число - "))

for i in range(1,k+1):

an=k\*i

print(k,'x',i,'=',an)

Можно использовать f-строку:

k=int(input("Введите число - "))

for i in range(1,k+1):

an=k\*i

print(f"{k} x {i} = {an}")

Рассмотрим еще несколько примеров использования цикла for для списка.

1.Ввести марку автомобиля, сравнить ее с марками списка и если марка найдена, то цикл завершаем с признаком found=True:

cars = ["Мерседес", "БМВ", "Ауди", "Тойота", "Форд"]

car = input("Введите марку автомобиля: ")

found = False

for car\_brand in cars:

if car == car\_brand:

found = True

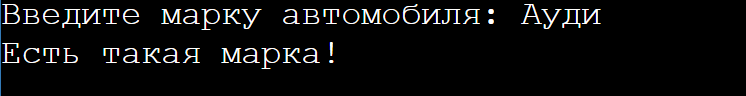
break

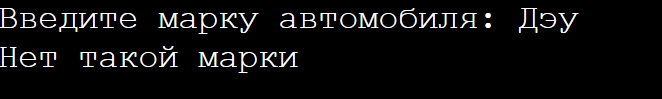
if found:

print("Есть такая марка!")

else:

print("Нет такой марки")





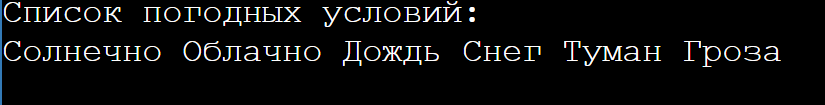
2.Выдача на экран списка в одну строку:

weather = ["Солнечно", "Облачно", "Дождь", "Снег", "Туман", "Гроза"]

print("Список погодных условий:")

for condition in weather:

print(condition, end=' ')



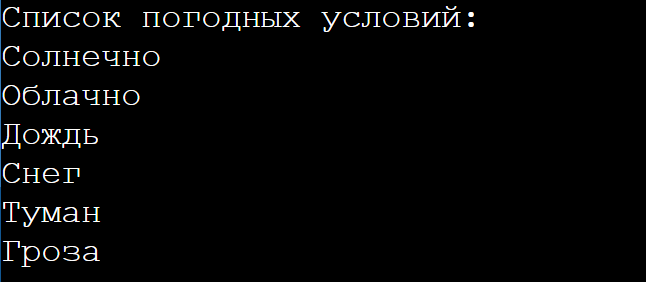
2.Выдача на экран списка в по-строчно:

weather = ["Солнечно", "Облачно", "Дождь", "Снег", "Туман", "Гроза"]

print("Список погодных условий:")

for condition in weather:

print(condition)



# Цикл while

**Цикл while** позволяет выполнить код неизвестное количество раз, пока выполняется некоторое условие. Цикл может быть выполнен 100 раз, единожды или вообще ни разу.

В цикле **while** условие проверяется до выполнения кода; это означает, что если при первой проверке условие не выполняется, цикл будет полностью пропущен. А значит, вы должны проследить за тем, чтобы перед запуском цикла были заданы правильные условия.

again = "yes"

while again == "yes":

print ("Привет")

again=input("Для продолжения введите yes - ")

Код будет повторяться до тех пор, пока пользователь не введет любой другой ответ, кроме «yes».

total = 0

while total < 100:

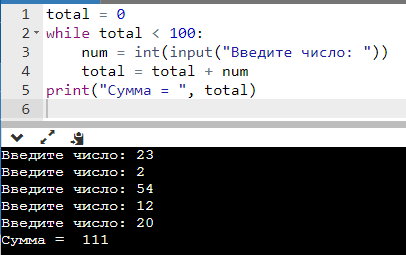
num = int(input("Введите число: "))

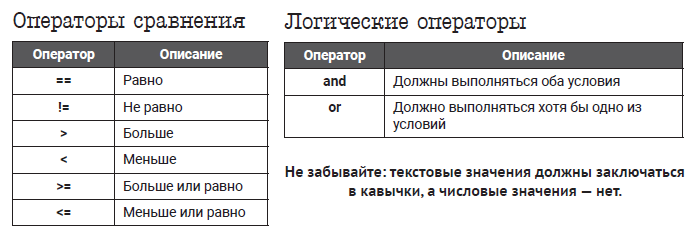
total = total + num

print("Сумма = ", total)

Эта программа создает переменную с именем **total** и сохраняет в ней значение 0. Она предлагает пользователю ввести число и прибавляет его к **total**. Цикл продолжает выполняться, пока значение **total** остается ниже 100. Как только **total** достигнет 100 и более, цикл перестанет выполняться и будет выведено значение **total**.

Далее приведен скриншот программы в online сервисе с результатом:





Рассмотрим еще несколько примеров применения цикла while при работе со **списком**.

1. Ввод названий улиц, дозапись в конец списка и выдача списка на экран:

streets = []

while True:

street = input("Введите название улиц (или введите 'end' для завершения): ")

if street == "end":

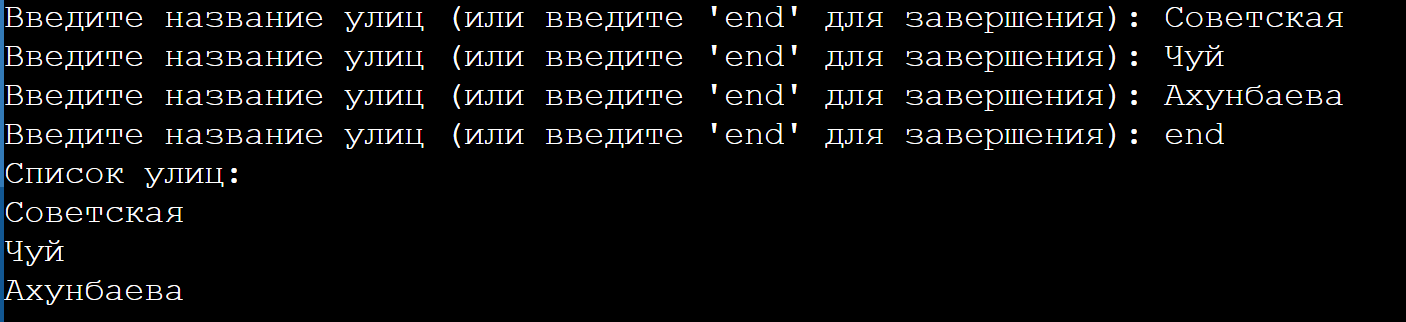
break

streets.append(street)

print("Список улиц:")

print(\*streets, sep="\n")

Результат:



2. Ввод количества улиц и в цикле ввод названий улиц, дозапись в конец списка и выдача списка на экран в одну строку:

streets = []

num\_ streets = int(input("Введите количество улиц: ")

count = 1

while count <= num\_ streets:

street = input(f"Введите название улицы {count}: ")

streets.append(street)

count += 1

print(\*streets)

# 

# Математические функции.

В Python при обработке данных можно применять различные математические функции, однако они доступны только в том случае, когда данные интерпретируются как целое число или как число с плавающей точкой (то есть число с дробной частью).

Если данные хранятся в виде строки, даже если эта строка содержит только цифровые символы, Python не сможет выполнять с ней вычисления.

Чтобы использовать некоторые математические функции (**math.sqrt(num)** и **math.pi**), необходимо импортировать библиотеку **math** в самом начале программы. Для этого достаточно ввести команду **import math** в первой

строке программы.

**print(round(num, 2))**

Выводит число, округленное до двух знаков в дробной части.

**\*\***

Выполняет возведение в степень (например, 102 записывается в виде **10\*\*2**).

**math.sqrt(num)**

Вычисляет квадратный корень из числа. Чтобы использовать эту функцию, необходимо включить строку **import math** в начало программы.

**num=float(input("Enter number: "))**

Позволяет использовать в вычислениях числа с плавающей точкой (то есть числа, имеющие как целую, так и дробную часть).

**math.pi**

Предоставляет значение числа «пи» (π) с точностью до 15 знаков. Чтобы использовать эту функцию, необходимо включить строку **import math** в начало программы.

**x // y**

Выполняет целочисленное деление (например, выражение **15//2** дает результат 7).

**x % y**

Вычисляет остаток (например, выражение **15%2** дает результат 1).

Конец лекции